What is claimed is:

1. A parallel counter comprising:

a plurality of inputs for receiving a binary number as a plurality of binary inputs;

a plurality of outputs for outputting binary code indicating the number of binary ones in the plurality of binary inputs; and

a logic circuit connected between the plurality of inputs and the plurality of binary outputs and for generating each of the plurality of binary outputs as an elementary symmetric function of the binary inputs.

2. A parallel counter according to claim 1, wherein said logic circuit is arranged to generate at least one of the binary outputs as an elementary symmetric function of the binary inputs using exclusive OR logic for combining a plurality of sets of one or more binary inputs.

3. A parallel counter according to claim 2, wherein said logic circuit is arranged to logically AND members of each set of binary inputs and to logically exclusively OR the result of the AND operations.

4. A parallel counter according to claim 3, wherein said logic circuit is arranged to logically AND $2^i$ of the binary inputs in each set for the generation of the $i^{th}$ binary output, where i is an integer from 1 to N, N is the number of binary outputs and i represents the significance of each binary output, each set being unique and the sets covering all possible combinations of binary inputs.

5. A parallel counter according to claim 3, wherein said logic circuit is arranged to logically AND members of each set of binary inputs, where each set is unique and the sets cover all possible combinations of binary inputs.

6. A parallel counter according to claim 1, wherein said logic circuit is arranged to generate at least one of the binary outputs as an elementary symmetric function of the binary inputs using OR logic for combining a plurality of sets of one or more binary inputs.

7. A parallel counter according to claim 6, wherein said logic circuit is arranged to logically AND members of each set of binary inputs and to logically OR the result of the AND operations.

8. A parallel counter according to claim 7, wherein said logic circuit is arranged to logically AND $2^{N-1}$ of the binary inputs in each set for the generation of the $N^{th}$ binary output, where N is the number of binary outputs and the $N^{th}$ binary output is the most significant, each set being unique and the sets covering all possible combinations of binary inputs.

9. A parallel counter according to claim 7, wherein said logic circuit is arranged to logically AND members of each set of binary inputs, where each set is unique and the sets cover all possible combinations of binary inputs.

10. A parallel counter according to claim 1, wherein said logic circuit is arranged to generate a first binary output as an elementary symmetric function of the binary inputs using exclusive OR logic for combining a plurality of sets of one or more binary inputs, and to generate an $N^{th}$ binary output as an elementary symmetric function of the binary inputs using OR logic for combining a plurality of sets of one or more binary inputs.

11. A parallel counter according to claim 1, wherein said logic circuit is arranged to generate two possible binary outputs for a binary output less significant than the $N^{th}$ binary output, as elementary symmetric functions of the binary inputs using OR logic for combining a plurality of sets of one or more binary inputs where N is the number of binary outputs, the sets used for each possible binary output being of two different sizes which are a function of the binary output being generated; and said logic circuit include selector logic to select one of the possible binary outputs based on a more significant binary output value.

12. A parallel counter according to claim 11, wherein said logic circuit is arranged to generate the two possible binary outputs for the $(N-1)^{th}$ binary output less significant

than the $N^{th}$ binary output, as elementary symmetric functions of the binary inputs using OR logic for combining a plurality of sets of one or more binary inputs, the sets used for each possible binary output being of size $2^{N-1} + 2^{N-2}$ and $2^{N-2}$ respectively and said selector logic being arranged to select one of the possible binary outputs based on the $N^{th}$ binary output value.

13. A parallel counter according to claim 1, wherein said logic circuit is arranged to generate two possible binary outputs for a binary output less significant than the $N^{th}$ binary output, as elementary symmetric functions of the binary inputs; and said logic circuit includes selector logic to select one of the possible binary outputs based on a more significant binary output value.

14. A parallel counter according to claim 1, wherein said logic circuit includes logic units for generating intermediate outputs as elementary symmetric functions of the binary inputs and is arranged to generate a binary output less significant than the $N^{th}$ binary output by combining intermediate outputs of the logic units by AND combining at least the intermediate output of one logic unit and an inverted output of another logic unit and OR combining the result of the AND combining with another intermediate output.

.

15. A parallel counter according to claim 14, wherein said logic circuit is arranged to generate the $k^{th}$ binary output $S_k$, where k=0 to t-1 and t is the number of outputs in accordance with the relationship:

$$S_k = \{OR\_n\_2^k \wedge \neg OR\_n\_2^{k+1}\} \vee \{OR\_n\_2^{k+1} + 2^k \wedge \neg OR\_n\_2^{k+2}\}$$
$$\vee \{OR\_n\_2^{k+2} + 2^k \wedge \neg OR\_n\_2^{k+2} + 2^{k+1}\}$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\vee OR\_n\_2^t + 2^{t-1} + 2^{t-2} + 2^k$$

where $\wedge$ is the logical AND operation, $\vee$ is the logical OR operation, and $\neg$ is an inversion operation.

16. A parallel counter according to claim 1, wherein said logic circuit includes a plurality of subcircuit logic modules each generating intermediate binary outputs as an elementary symmetric function of some of the binary inputs, and logic for logically combining the intermediate binary outputs to generate said binary outputs.

17. A parallel counter according to claim 16, wherein said subcircuit logic modules are arranged to use OR logic for combining sets of said some of said binary inputs.

18. A parallel counter according to claim 17, wherein said logic circuit includes one or more logic modules each for generating a binary output as an elementary symmetric function of the binary inputs using executive OR logic for combining a plurality of sets of one or more binary inputs.

19. A parallel counter according to claim 1, wherein said logic circuit implements a large elementary symmetric function by implementing a plurality of small elementary symmetric functions and combining the results.

20. A parallel counter according to claim 1, wherein said logic circuit is divided into a plurality of logic units, each logic unit is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to the logic unit, the binary inputs of said plurality of inputs are divided into inputs into a plurality of said logic units, and the binary outputs of said plurality of outputs are generated using binary outputs of a plurality of said logic units.

21. A parallel counter according to claim 20, wherein the logic units are arranged hierarchically such that logic units at a higher level in the hierarchy include the logic of at least one logic unit at a lower level in the hierarchy and have more of the binary inputs as inputs than the logic units at a lower level in the hierarchy

22. A parallel counter according to claim 20, wherein the binary inputs of said plurality of inputs are divided according to a binary tree into inputs into a plurality of said logic units.

23. A parallel counter according to claim 22, wherein said logic units are arranged to receive $2^n$ of said binary inputs, where n is an integer indicating the level of the logic units in the binary tree, said logic circuit has m logic units at each level, where m is a rounded up integer determined from (the number of binary inputs)/ $2^n$, logic units having a higher level in the binary tree comprise logic of logic units at lower levels in the binary tree, and each logic unit is arranged to generate logic unit binary outputs as a symmetric function of the binary inputs to the logic unit.

24. A parallel counter according to claim 23, wherein each logic unit at the first level is arranged to generate logic unit binary outputs as a small elementary symmetric function of the binary inputs to said logic circuit.

25. A parallel counter according to claim 23, wherein each logic unit at the first level is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to said logic circuit using OR logic for combining the binary inputs.

26. A parallel counter according to claim 25, wherein each logic unit at the first level is arranged to logically AND each of the binary inputs to the logic unit and to logically OR each of the binary inputs to the logic unit to generate the logic unit binary outputs.

27. A parallel counter according to claim 24, wherein each logic unit at the first level is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to said logic circuit using exclusive OR logic for combining the binary inputs.

28. A parallel counter according to claim 27, wherein each logic unit at the first level is arranged to logically AND each of the binary inputs to the logic unit and to logically exclusively OR each of the binary inputs to the logic unit to generate the logic unit binary outputs.

29. A parallel counter according to claim 23, wherein elementary logic units are provided as the logic units at the first level for performing elementary symmetric functions, outputs from each of two primary elementary logic units receiving four logically adjacent binary inputs from said plurality of inputs are input to two secondary elementary logic units, an output from each of the secondary elementary logic units is input to a tertiary elementary logic unit, and said primary, secondary and tertiary elementary logic units form a secondary logic unit at a second level of the binary tree having a binary output comprising a binary output from each of said secondary elementary logic units and two binary outputs from said tertiary elementary logic unit.

30. A parallel counter according to claim 29, wherein tertiary logic units at a third level of the binary tree each comprise two secondary logic units receiving eight logically adjacent binary inputs from said plurality of inputs, four elementary logic units receiving as inputs the outputs of said two secondary logic units, and further logic for generating binary outputs as a symmetric function of the binary inputs to said tertiary logic unit using the binary outputs of said four elementary logic units.

31. A parallel counter according to claim 30, wherein quaternary logic units at a fourth level of the binary tree each comprise two tertiary logic units receiving sixteen logically adjacent binary inputs from said plurality of inputs, four elementary logic units receiving as inputs the outputs of said two tertiary logic units, and further logic for generating binary outputs as a symmetric function of the binary inputs to said quaternary logic unit using the binary outputs of said four elementary logic units

32. A parallel counter according to claim 23, wherein elementary logic units are provided as the logic units at the first level for performing elementary symmetric functions, and logic units for higher levels are comprised of logic units of lower levels.

33. A parallel counter according to claim 32, wherein said logic units for higher levels above the second level comprise logic units of an immediately preceding level and elementary logic units.

34. A parallel counter according to claim 23, wherein each logic unit at each level is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to said logic circuit using OR logic for combining the binary inputs.

35. A parallel counter according to claim 23, wherein each logic unit at each level is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to said logic circuit using exclusive OR logic for combining the binary inputs.

36. A parallel counter according to claim 20, wherein the logic units are arranged hierarchically and at least one logic unit in at least one level of the hierarchy implements an inverted elementary symmetric function.

37. A parallel counter according to claim 36, wherein logic units at an odd number of levels in the hierarchy implement inverted elementary symmetric functions, logic units at an even number of levels in the hierarchy implement symmetric functions, and the inputs to the logic units at the first level of the hierarchy are inverted.

38. A parallel counter according to claim 36, wherein logic units at an even number of levels in the hierarchy implement inverted elementary symmetric functions, logic units at an even number of levels in the hierarchy implement symmetric functions, and the inputs to the logic units at the first level of the hierarchy are input to logic units in a first level in the hierarchy uninverted.

39. A parallel counter according to claim 36, wherein logic units at an even number of levels in the hierarchy implement inverted elementary symmetric functions, logic units at an odd number of levels in the hierarchy implement symmetric functions, and the inputs to the logic units at the first level of the hierarchy are inverted.

40. A parallel counter according to claim 36, wherein at least one logic unit in at least one level of the hierarchy implements an elementary symmetric function and the or

each inverted elementary symmetric function and the or each elementary symmetric function are implemented in alternated levels in the hierarchy.

41. A parallel counter according to claim 36, wherein logic units in at least one level in the hierarchy comprise inversion logic.

42. A parallel counter according to claim 36, wherein the logic units are arranged hierarchically in a binary tree structure.

43. A parallel counter according to claim 1, wherein the number of inputs is at least four and the number of outputs is at least three.

44. A parallel counter comprising:

at least five inputs for receiving a binary number as a plurality of binary inputs;

at least three outputs for outputting binary code indicating the number of binary ones in the plurality of binary inputs; and

a logic circuit connected between the plurality of inputs and the plurality of binary outputs and for generating each of the plurality of binary outputs as an elementary symmetric function of the binary inputs.

45. A parallel counter according to claim 44, wherein said logic circuit is arranged to generate at least two outputs independently of each other.

46. A parallel counter comprising:

n inputs for receiving a binary number as binary inputs, where $4 \geq n \geq 7$;

three outputs for outputting binary code indicating the number of binary ones in the binary inputs; and

a logic circuit connected between the inputs and the three outputs and for generating a first output as an elementary symmetric function EXOR_n_1 of the binary inputs, a second output as a combination of three elementary symmetric functions

OR_n_2, OR_n_4 and OR_n_6, and a third output as an elementary symmetric function OR_n_4.

47. A parallel counter comprising:

n inputs for receiving a binary number as binary inputs, where $8 \geq n \geq 15$;

four outputs for outputting binary code indicating the number of binary ones in the binary inputs; and

a logic circuit connected between the inputs and the four outputs and for generating a first output as an elementary symmetric function EXOR_n_1 of the binary inputs, a second output as an elementary symmetric function EXOR_n_2 of the binary inputs, a third output as a combination of three elementary symmetric functions OR_n_4_, OR_n_8 and OR_n_12, and a third output as an elementary symmetric function OR_n_8.

48. A conditional parallel counter having m possible high inputs out of n inputs, where $m<n$, and n and m are integers, the conditional parallel counter comprising the parallel counter according to claim 1 for counting inputs to generate p outputs for m inputs, wherein the number n of inputs to the counter is greater than $2^p$.

49. A constant multiplier comprising the conditional parallel counter according to claim 48.

50. A digital filter comprising a constant multiplier according to claim 48.

51. A logic circuit including the parallel counter according to claim 1.

52. An integrated circuit including the parallel counter according to claim 1.

53. A digital electronic device including the parallel counter according to claim 1.

54. A logic circuit for multiplying two binary numbers comprising:

array generation logic for generating an array of binary numbers comprising combinations of each bit of each binary number;

array reduction logic including at least one parallel counter according to claim 1 for reducing the number of combinations in the array; and

binary addition logic for adding the reduced combinations to generate an output.

55. A logic circuit for multiplying two binary numbers, the logic circuit comprising:

array generation logic for generating, from the two binary numbers, an array of binary values which are required to be added, and for further logically combining binary values in the array to generate the array in which the maximal depth of the array is below N bits, where N is the number of bits of the largest of the two binary numbers;

array reduction logic for reducing the depth of the array to two binary numbers; and

addition logic for adding the binary values of the two binary numbers.

56. A logic circuit according to claim 55, wherein said array generation logic is arranged to perform a logical binary operation between each bit in one binary number and each bit in the other binary number to generate an array of logical binary combinations comprising an array of binary values.

57. A logic circuit according to claim 56, wherein said array generation logic is arranged to perform a logical AND operation between each bit in one binary number and each bit in the other binary number to generate an array of logical AND combinations comprising an array of binary values.

58. A logic circuit according to claim 57, wherein said array generation logic is arranged to perform the further logical combination of values for values formed by the logical binary combination of each bit $A_i$ of one binary number and each bit $B_j$ of the other binary number, where $i-j-k \leq 1$, k is a chosen integer, and i and j are integers from 1 to N.

59. A logic circuit according to claim 55, wherein said array generation logic is arranged to logically AND combine each bit $A_i$ of the first binary number with each bit $B_j$ of a second binary number to generate said array comprising a sequence of binary numbers represented by said logical AND combinations, $A_i$ AND $B_j$ and to carry out further logical combination by logically combining the combination $A_1$ AND $B_{N-2}$, $A_1$ AND $B_{N-}1$ where N is the number of bits in the binary numbers.

60. A logic circuit according to claim 59, wherein said array generation logic is arranged to combine the combinations $A_1$ AND $B_{N-2}$ and $A_0$ AND $B_{N-1}$, using exclusive OR logic to replace these combinations, and to combine $A_1$ AND $B_{N-1}$ and $A_0$ AND $B_{N-2}$ to replace the $A_1$ AND $B_{N-1}$ combination.

61. A logic circuit according to claim 55, wherein said array reduction logic includes at least one of: at least one full adder, at least one half adder, and at least one parallel counter.

62. A logic circuit according to claim 61, wherein said array reduction logic includes at least one parallel counter according to claim 1.

63. A multiply-accumulate logic circuit comprising the logic circuit according to claim 55, wherein said array generation logic is arranged to include an accumulation of previous multiplications.

64. An integrated circuit including the logic circuit according to claim 55.

65. A digital electronic device including the logic circuit according to claim 55.

66. A logic circuit comprising:
at least four inputs for receiving a binary number as a plurality of binary inputs;
at least one output for outputting binary code; and

logic elements connected between the plurality of inputs and the or each binary output and for generating the or each binary output in accordance with a threshold function implemented as a binary tree and having a threshold of at least 2.

67. A logic circuit according to claim 66, wherein the logic elements are arranged to generate the or each binary output as an elementary symmetric function of the binary inputs.

68. A logic circuit according to claim 67, wherein the logic elements are arranged to generate at least one of the binary outputs as an OR symmetric function of the binary inputs.

69. A logic circuit according to claim 68, wherein the logic elements are arranged to generate at least one of the binary outputs as an exclusive OR symmetric function of the binary inputs.

70. A logic circuit according to claim 66, wherein said logic elements comprise a plurality of subcircuit logic modules each generating intermediate binary outputs as an elementary symmetric function of some of the binary inputs, and logic for logically combining the intermediate binary outputs to generate the or each binary output.

71. A logic circuit according to claim 67, wherein said logic elements comprise a plurality of logic modules each for generating intermediate binary outputs as an elementary symmetric function of some of the binary inputs, and logic for logically combining the intermediate binary outputs to generate the or each binary output, and.the logic modules are arranged hierarchically and at least one logic module in at least one level of the hierarchy implements an inverted elementary symmetric function.

72. A logic circuit according to claim 71, wherein logic modules at an odd number of levels in the hierarchy implement inverted elementary symmetric functions, logic modules at an even number of levels in the hierarchy implement symmetric functions, and the inputs to the logic modules at the first level of the hierarchy are inverted.

73. A logic circuit according to claim 71, wherein logic modules at an even number of levels in the hierarchy implement inverted elementary symmetric functions, logic modules at an even number of levels in the hierarchy implement symmetric functions, and the inputs to the logic modules at the first level of the hierarchy are input to logic units in a first level in the hierarchy uninverted.

74. A logic circuit according to claim 71, wherein logic modules at an even number of levels in the hierarchy implement inverted elementary symmetric functions, logic modules at an odd number of levels in the hierarchy implement symmetric functions, and the inputs to the logic modules at the first level of the hierarchy are inverted.

75. A logic circuit according to claim 71, wherein at least one logic module in at least one level of the hierarchy implements an elementary symmetric function, and the or each inverted elementary symmetric function and the or each elementary symmetric function are implemented in alternated levels in the hierarchy.

76. A logic circuit according to claim 71, wherein logic modules in at least one level in the hierarchy comprise inversion logic.

77. A logic circuit according to claim 71, wherein the logic modules are arranged hierarchically in a binary tree structure.

78. A logic circuit comprising:

at least four inputs for receiving a binary number as a plurality of binary inputs;

at least one output for outputting binary code; and

logic elements connected between the plurality of inputs and the plurality of binary outputs arranged to generate the or each of the plurality of binary outputs as an elementary symmetric function of the binary inputs.

79. A logic circuit according to claim 78, wherein said logic elements are arranged to generate at least one of the binary outputs as an elementary symmetric function of the

binary inputs using exclusive OR logic for combining a plurality of sets of one or more binary inputs.

80. A logic circuit according to claim 79, wherein said logic elements are arranged to logically AND members of each set of binary inputs and to logically exclusively OR the result of the AND operations.

81. A logic circuit according to claim 80, wherein said logic elements are arranged to logically AND $2^i$ of the binary inputs in each set for the generation of the $i^{th}$ binary output, where i is an integer from 1 to N, N is the number of binary outputs and i represents the significance of each binary output, each set being unique and the sets covering all possible combinations of binary inputs.

82. A logic circuit according to claim 80, wherein said logic elements are arranged to logically AND members of each set of binary inputs, where each set is unique and the sets cover all possible combinations of binary inputs.

83. A logic circuit according to claim 78, wherein said logic elements are arranged to generate at least one of the binary outputs as an elementary symmetric function of the binary inputs using OR logic for combining a plurality of sets of one or more binary inputs.

84. A logic circuit according to claim 83, wherein said logic elements are arranged to logically AND members of each set of binary inputs and to logically OR the result of the AND operations.

85. A logic circuit according to claim 84, wherein said logic elements are arranged to logically AND $2^{N-1}$ of the binary inputs in each set for the generation of the $N^{th}$ binary output, where N is the number of binary outputs and the $N^{th}$ binary output is the most significant, each set being unique and the sets covering all possible combinations of binary inputs.

86. A logic circuit according to claim 84, wherein said logic elements are arranged to logically AND members of each set of binary inputs, where each set is unique and the sets cover all possible combinations of binary inputs.

87. A logic circuit according to claim 78, wherein said logic elements are arranged to generate a first binary output as an elementary symmetric function of the binary inputs using exclusive OR logic for combining a plurality of sets of one or more binary inputs, and to generate an $N^{th}$ binary output as an elementary symmetric function of the binary inputs using OR logic for combining a plurality of sets of one or more binary inputs.

88. A logic circuit according to claim 78, wherein said logic elements are arranged to generate two possible binary outputs for a binary output less significant than the $N^{th}$ binary output, as elementary symmetric functions of the binary inputs using OR logic for combining a plurality of sets of one or more binary inputs where N is the number of binary outputs, the sets used for each possible binary output being of two different sizes which are a function of the binary output being generated; and said logic elements include selector logic to select one of the possible binary outputs based on a more significant binary output value.

89. A logic circuit according to claim 88, wherein said logic elements are arranged to generate the two possible binary outputs for the $(N-1)^{th}$ binary output less significant than the $N^{th}$ binary output, as elementary symmetric functions of the binary inputs using OR logic for combining a plurality of sets of one or more binary inputs, the sets used for each possible binary output being of size $2^{N-1} + 2^{N-2}$ and $2^{N-2}$ respectively and said selector logic being arranged to select one of the possible binary outputs based on the $N^{th}$ binary output value.

90. A logic circuit according to claim 78, wherein said logic elements are arranged to generate two possible binary outputs for a binary output less significant than the $N^{th}$ binary output, as elementary symmetric functions of the binary inputs; and said logic

elements include selector logic to select one of the possible binary outputs based on a more significant binary output value.

91. A logic circuit according to claim 78, wherein said logic elements include logic units for generating intermediate outputs as elementary symmetric functions of the binary inputs, and are arranged to generate a binary output less significant than the $N^{th}$ binary output by combining intermediate outputs of the logic units by AND combining at least the intermediate output of one logic unit and an inverted output of another logic unit and OR combining the result of the AND combining with another intermediate output.

92. A logic circuit according to claim 91, wherein said logic elements are arranged to generate the $k^{th}$ binary output $S_k$, where k=0 to t-1 and t is the number of outputs in accordance with the relationship:

$$S_k = \{OR\_n\_2^k \wedge \neg OR\_n\_2^{k+1}\} \vee \{OR\_n\_2^{k+1} + 2^k \wedge \neg OR\_n\_2^{k+2}\}$$
$$\vee \{OR\_n\_2^{k+2} + 2^k \wedge \neg OR\_n\_2^{k+2} + 2^{k+1}\}$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\vee OR\_n\_2^t + 2^{t-1} + 2^{t-2} + 2^k$$

where $\wedge$ is the logical AND operation, $\vee$ is the logical OR operation, and $\neg$ is an inversion operation.

93. A logic circuit according to claim 78, wherein said logic elements include a plurality of subcircuit logic modules each generating intermediate binary outputs as an elementary symmetric function of some of the binary inputs, and logic for logically combining the intermediate binary outputs to generate said binary outputs.

94. A logic circuit according to claim 93, wherein said subcircuit logic modules are arranged to use OR logic for combining sets of said some of said binary inputs.

95. A logic circuit according to claim 94, wherein said logic elements include one or more logic modules each for generating a binary output as an elementary symmetric

function of the binary inputs using executive OR logic for combining a plurality of sets of one or more binary inputs.

96. A logic circuit according to claim 78, wherein said logic elements implement a large elementary symmetric function by implementing a plurality of small elementary symmetric functions and combining the results.

97. A logic circuit according to claim 78, wherein said logic elements are divided into a plurality of logic units, each logic unit is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to the logic unit, the binary inputs of said plurality of inputs are divided into inputs into a plurality of said logic units, and the binary outputs of said plurality of outputs are generated using binary outputs of a plurality of said logic units.

98. A logic circuit according to claim 97, wherein the logic units are arranged hierarchically such that logic units at a higher level in the hierarchy include the logic of at least one logic unit at a lower level in the hierarchy and have more of the binary inputs as inputs than the logic units at a lower level in the hierarchy

99. A logic circuit according to claim 97, wherein the binary inputs of said plurality of inputs are divided according to a binary tree into inputs into a plurality of said logic units.

100. A logic circuit according to claim 99, wherein said logic units are arranged to receive $2^n$ of said binary inputs, where n is an integer indicating the level of the logic units in the binary tree, said logic circuit has m logic units at each level, where m is a rounded up integer determined from (the number of binary inputs)/ $2^n$, logic units having a higher level in the binary tree comprise logic of logic units at lower levels in the binary tree, and each logic unit is arranged to generate logic unit binary outputs as a symmetric function of the binary inputs to the logic unit.

101. A logic circuit according to claim 100, wherein each logic unit at the first level is arranged to generate logic unit binary outputs as a small elementary symmetric function of the binary inputs to said logic circuit.

102. A logic circuit according to claim 100, wherein each logic unit at the first level is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to said logic circuit using OR logic for combining the binary inputs.

103. A logic circuit according to claim 102, wherein each logic unit at the first level is arranged to logically AND each of the binary inputs to the logic unit and to logically OR each of the binary inputs to the logic unit to generate the logic unit binary outputs.

104. A logic circuit according to claim 101, wherein each logic unit at the first level is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to said logic circuit using exclusive OR logic for combining the binary inputs.

105. A logic circuit according to claim 104, wherein each logic unit at the first level is arranged to logically AND each of the binary inputs to the logic unit and to logically exclusively OR each of the binary inputs to the logic unit to generate the logic unit binary outputs.

106. A logic circuit according claim 100, wherein elementary logic units are provided as the logic units at the first level for performing elementary symmetric functions, outputs from each of two primary elementary logic units receiving four logically adjacent binary inputs from said plurality of inputs are input to two secondary elementary logic units, an output from each of the secondary elementary logic units is input to a tertiary elementary logic unit, and said primary, secondary and tertiary elementary logic units form a secondary logic unit at a second level of the binary tree having a binary output comprising a binary output from each of said secondary elementary logic units and two binary outputs from said tertiary elementary logic unit.

107. A logic circuit according to claim 106, wherein tertiary logic units at a third level of the binary tree each comprise two secondary logic units receiving eight logically adjacent binary inputs from said plurality of inputs, four elementary logic units receiving as inputs the outputs of said two secondary logic units, and further logic for generating binary outputs as a symmetric function of the binary inputs to said tertiary logic unit using the binary outputs of said four elementary logic units.

108. A logic circuit according to claim 107, wherein quaternary logic units at a fourth level of the binary tree each comprise two tertiary logic units receiving sixteen logically adjacent binary inputs from said plurality of inputs, four elementary logic units receiving as inputs the outputs of said two tertiary logic units, and further logic for generating binary outputs as a symmetric function of the binary inputs to said quaternary logic unit using the binary outputs of said four elementary logic units

109. A logic circuit according to claim 100, wherein elementary logic units are provided as the logic units at the first level for performing elementary symmetric functions, and logic units for higher levels are comprised of logic units of lower levels.

110. A logic circuit according to claim 109, wherein said logic units for higher levels above the second level comprise logic units of an immediately preceding level and elementary logic units.

111. A logic circuit according to claim 100, wherein each logic unit at each level is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to said logic circuit using OR logic for combining the binary inputs.

112. A logic circuit according to claim 100, wherein each logic unit at each level is arranged to generate logic unit binary outputs as an elementary symmetric function of the binary inputs to said logic circuit using exclusive OR logic for combining the binary inputs.

113. A logic circuit according to claim 97, wherein the logic units are arranged hierarchically and at least one logic unit in at least one level of the hierarchy implements an inverted elementary symmetric function.

114. A logic circuit according to claim 113, wherein logic units at an odd number of levels in the hierarchy implement inverted elementary symmetric functions, logic units at an even number of levels in the hierarchy implement symmetric functions, and the inputs to the logic units at the first level of the hierarchy are inverted.

115. A logic circuit according to claim 113, wherein logic units at an even number of levels in the hierarchy implement inverted elementary symmetric functions, logic units at an even number of levels in the hierarchy implement symmetric functions, and the inputs to the logic units at the first level of the hierarchy are input to logic units in a first level in the hierarchy uninverted.

116. A logic circuit according to claim 113, wherein logic units at an even number of levels in the hierarchy implement inverted elementary symmetric functions, logic units at an odd number of levels in the hierarchy implement symmetric functions, and the inputs to the logic units at the first level of the hierarchy are inverted.

117. A logic circuit according to claim 113, wherein at least one logic unit in at least one level of the hierarchy implements an elementary symmetric function and the or each inverted elementary symmetric function and the or each elementary symmetric function are implemented in alternated levels in the hierarchy.

118. A logic circuit according to claim 113, wherein logic units in at least one level in the hierarchy comprise inversion logic.

119. A logic circuit according to claim 113, wherein the logic units are arranged hierarchically in a binary tree structure.

120. An integrated circuit including the logic circuit according to claim 78.

121. A digital electronic device including the logic circuit according to claim 78.

122. A logic circuit for multiplying two binary numbers comprising:

array generation logic for generating an array of binary numbers comprising combinations of each bit of each binary number;

array reduction logic including at least one logic circuit according to claim 64 for reducing the number of combinations in the array; and

binary addition logic for adding the reduced combinations to generate an output.

123. A method of designing a logic circuit comprising a plurality of inputs for receiving a binary number as a plurality of binary inputs, at least one output for outputting binary code, and logic elements connected between the plurality of inputs and the or each binary output and arranged to generate the or each binary output as a threshold function of the binary inputs, the method comprising:

determining logic elements for performing the threshold functions; and

reducing the logic elements by identifying logic elements performing a logical AND of two threshold functions and reducing the identified logic elements to logic elements for performing the threshold function having the higher threshold, and identifying logic elements performing a logical OR of two threshold functions and reducing the identified logic elements to logic elements for performing the threshold function having the lower threshold.

124. A method according to claim 123, wherein the reduction is performed using logical OR threshold functions having the relationship:

$$OR\_n\_k \wedge OR\_n\_s = OR\_n\_k$$

$$OR\_n\_k \vee OR\_n\_s = OR\_n\_s$$

where $k \geq s$, n is the number of inputs and k and s are the number of high inputs.

125. A method according to claim 123, wherein the logic elements are designed for performing the threshold functions as elementary symmetric functions, and the logic

circuit is designed to generate the or each binary output as an elementary symmetric function.

126. A system for designing a logic circuit comprising a plurality of inputs for receiving a binary number as a plurality of binary inputs, at least one output for outputting binary code, and logic elements connected between the plurality of inputs and the or each binary output and arranged to generate the or each binary output as a threshold function of the binary inputs, the system comprising:

determining means for determining logic elements for performing the threshold functions; and

reducing means for reducing the logic elements by identifying logic elements performing a logical AND of two threshold functions and reducing the identified logic elements to logic elements for performing the threshold function having the higher threshold, and identifying logic elements performing a logical OR of two threshold functions and reducing the identified logic elements to logic elements for performing the threshold function having the lower threshold.

127. A system according to claim 126, wherein said reducing means is adapted to perform the reduction using logical OR threshold functions having the relationship:

$$OR\_n\_k \wedge OR\_n\_s = OR\_n\_k$$

$$OR\_n\_k \vee OR\_n\_s = OR\_n\_s$$

where $k \geq s$, n is the number of inputs and k and s are the number of high inputs.

128. A system according to claim 126, wherein said determining means is adapted to design the logic elements to perform the threshold functions as elementary symmetric functions, and to generate the or each binary output as an elementary symmetric function.

129. A computer system for designing a logic circuit comprising a plurality of inputs for receiving a binary number as a plurality of binary inputs, at least one output for outputting binary code, and logic elements connected between the plurality of inputs

and the or each binary output and arranged to generate the or each binary output as a threshold function of the binary inputs, the computer system comprising:

a memory storing computer readable code;

a processor for reading and implementing the code;

wherein the code stored in the memory comprises code for controlling the processor to:

determine logic elements for performing the threshold functions; and

reduce the logic elements by identifying logic elements performing a logical AND of two threshold functions and reducing the identified logic elements to logic elements for performing the threshold function having the higher threshold, and identifying logic elements performing a logical OR of two threshold functions and reducing the identified logic elements to logic elements for performing the threshold function having the lower threshold.

130. A computer system according to claim 129, wherein the code stored in the memory comprises code for controlling the processor to: perform the reduction using logical OR threshold functions having the relationship:

$$OR\_n\_k \wedge OR\_n\_s = OR\_n\_k$$

$$OR\_n\_k \vee OR\_n\_s = OR\_n\_s$$

where $k \geq s$, n is the number of inputs and k and s are the number of high inputs.

131. A computer system according to claim 129, wherein the code stored in the memory comprises code for controlling the processor to: design the logic elements to perform the threshold functions as elementary symmetric functions, and to generate the or each binary output as an elementary symmetric function.

132. A carrier medium carrying computer readable instructions for controlling a computer to implement the method of claim 123.

133. A method of designing a logic circuit comprising a plurality of inputs for receiving a binary number as a plurality of binary inputs, at least one output for

outputting binary code, and logic elements connected between the plurality of inputs and the binary outputs and arranged to generate each binary output as a symmetric function of the binary inputs, the method comprising:

designing the logic circuit using exclusive OR logic;

identifying any logic which cannot have inputs that are high at the same time; and

replacing the identified exclusive OR logic with OR logic.

134. A method according to claim 133, wherein the logic circuit is designed to generate each binary output as an elementary symmetric function of the binary inputs.

135. A method according to claim 133, wherein the logic circuit is designed as a parallel counter having a plurality of outputs.

136. A system for designing a logic circuit comprising a plurality of inputs for receiving a binary number as a plurality of binary inputs, a plurality of outputs for outputting binary code, and logic elements connected between the plurality of inputs and the binary outputs and arranged to generate each binary output as a symmetric function of the binary inputs, the system comprising:

designing means for designing the logic circuit using exclusive OR logic;

identifying means for identifying any logic which cannot have inputs that are high at the same time; and

replacing means for replacing the identified exclusive OR logic with OR logic.

137. A system according to claim 136, wherein said designing means is adapted to design the logic circuit to generate each binary output as an elementary symmetric function of the binary inputs.

138. A system according to claim 136, wherein said designing means is adapted to design the logic circuit as a parallel counter having a plurality of outputs.

139. A computer system for designing a logic circuit comprising a plurality of inputs for receiving a binary number as a plurality of binary inputs, a plurality of outputs for outputting binary code, and logic elements connected between the plurality of inputs and the binary outputs and arranged to generate each binary output as a symmetric function of the binary inputs, the system comprising:

a memory storing computer readable code;

a processor for reading and implementing the code;

wherein the code stored in the memory comprises code for controlling the processor to:

design the logic circuit using exclusive OR logic;

identify any logic which cannot have inputs that are high at the same time; and

replace the identified exclusive OR logic with OR logic.

140. A computer system according to claim 139, wherein the code stored in the memory comprises code for controlling the processor to design the logic circuit to generate each binary output as an elementary symmetric function of the binary inputs.

141. A computer system according to claim 139, wherein the code stored in the memory comprises code for controlling the processor to design the logic circuit as a parallel counter having a plurality of outputs.

142. A carrier medium carrying computer readable instructions for controlling a computer to implement the method of claim 133.

143. A method of designing a logic circuit comprising:

providing a library of logic module designs each for performing a small symmetric function;

designing a logic circuit to perform a large symmetric function;

identifying small symmetric functions which can perform said symmetric function;

selecting logic modules from said library to perform said small symmetric functions;

identifying a logic circuit in the selected logic circuit which performs a symmetric function and which can be used to perform another symmetric function; and

selecting the logic circuit corresponding to the identified symmetric function and using the selected logic circuit with inverters to perform said other symmetric function using the relationship between the symmetric functions:

$$OR\_n\_k(X_1...X_n) = \neg OR\_n\_(n+1-k)(\neg X_1...\neg X_n)$$

where $\neg$ denotes an inversion, n is the number of inputs, and k is the number of sets of inputs AND combined together.

144. A method according to claim 143, wherein the symmetric functions are elementary symmetric functions.

145. A system for designing a logic circuit comprising:

storing means storing a library of logic module designs each for performing a small symmetric function;

designing means for designing a logic circuit to perform a large symmetric function;

identifying small symmetric functions which can perform said symmetric function;

first selecting means for selecting logic modules from said library to perform said small symmetric functions;

identifying means for identifying a logic circuit in the selected logic circuit which performs a symmetric function and which can be used to perform another symmetric function; and

second selecting means for selecting the logic circuit corresponding to the identified symmetric function and using the selected logic circuit with inverters to perform said other symmetric function using the relationship between the symmetric functions:

$$OR\_n\_k(X_1...X_n) = \neg OR\_n\_(n+1-k)(\neg X_1...\neg X_n)$$

where $\neg$ denotes an inversion, n is the number of inputs, and k is the number of sets of inputs AND combined together.

146. A system according to claim 145, wherein the symmetric functions are elementary symmetric functions.

147. A computer system for designing a logic circuit comprising:

a data memory storing a library of logic module designs each for performing a small symmetric function;

a code memory storing computer readable code;

a processor for reading and implementing the code;

wherein the code stored in the code memory comprises code for controlling the processor to:

design a logic circuit to perform a large symmetric function;

identifying small symmetric functions which can perform said symmetric function;

select logic modules from said library to perform said small symmetric functions;

identify a logic circuit in the selected logic circuit which performs a symmetric function and which can be used to perform another symmetric function; and

select the logic circuit corresponding to the identified symmetric function and using the selected logic circuit with inverters to perform said other symmetric function using the relationship between the symmetric functions:

$$OR\_n\_k(X_1...X_n) = \neg OR\_n\_(n+1-k)(\neg X_1...\neg X_n)$$

where $\neg$ denotes an inversion, n is the number of inputs, and k is the number of sets of inputs AND combined together.

148. A computer system according to claim 147, wherein the symmetric functions are elementary symmetric functions.

149. A carrier medium carrying computer readable instructions for controlling a computer to implement the method of claim 143.